INTERSPEECH 2016: Show & Tell Contribution
September 8–12, 2016, San Francisco, USA

A Low Cost Desktop Robot and Tele-presence Device for Interactive Speech Research

Michael C. Brady
Department of Computer Science, Tufts University
mcbrady@tufts.edu

Abstract

In building robotic systems that interact with people through speech, many robotics engineers are obliged to treat artificial speech recognition and synthesis as a black-box problem best left to speech engineers to solve. Yet speech engineers today typically do not have access to the kinds of expensive robots needed for this development. Progress on the human-robot speech interface thus suffers from something of a diffusion of responsibility. In an attempt to remedy the situation, we have developed a low-cost interactive embodied speech device. The device is constructed from off-the-shelf components and from 3D-printed and laser-cut parts. We make the files for the 3D and laser-cut parts freely available for download. In addition to offering basic assembled devices and kits for self-assembly, we provide an assembly guide and a shopping list of components a user will need in order to build, maintain, and customize their own device. We supply a basic software framework (in both Matlab and in C/C++), and template code for a ROS node for interfacing with the device. The idea is to establish a standard and accessible hardware platform with an open-source foundation for the sharing of ideas and research.

Index Terms: multi-modal speech recognition and synthesis, human-robot interaction, open-source speech software.

1. Why?

It is strategic to develop a speech system that takes advantage of all available perceptual cues so that effective and efficient communication with the robot is achieved. The robot may use visual information (face and gesture recognition) and active perception (such as moving its head around to help in mapping the acoustic environment) in addition to performing non-verbal communicative acts (such as shifting its apparent gaze to an object for shared attention). Speech communication with robots involves more than this however, and involves more than perhaps producing empathetic and companion-oriented artificial agent behaviors. With robots, an important goal of speech communication is to instruct the robot to do something. A robot must reason about what the commanded action is, how to achieve the action, and the potential consequences of the action (will the action break something or hurt someone?). The task management and knowledge representation systems of robot control architectures tend to be beyond what speech systems typically involve. Feedback from the robot cognitive system needs to be incorporated into the speech interface. It is therefore strategic to provide a common and readily-accessible platform upon which speech engineers and robotics engineers (and hobbyists) may work together to share ideas and imagine new ways to solve problems.

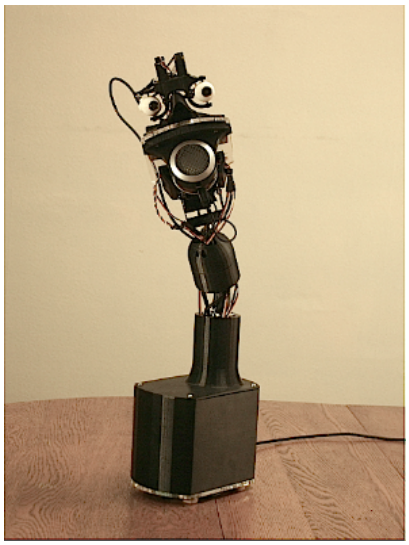

Figure 1: *minimal desktop interactive speech device*

2. Hardware

Two versions of the system are presented. One is a minimal or starter system, as seen in Figure 1 (total cost for parts is below that of a laptop computer), and the other is a more elaborate system with arms and a Kinect V2 sensor, as seen in Figure 2. Each device is built from parts that are available for sale over the internet and in local stores. These parts include a motor controller (we are using the SSC32, sold by Lynxmotion.com [1]), standard hobby servo motors, and erector-set style components. The eye-cams use USB endoscope cameras that are on the same tilt motor but on separate pan motors so that the robot may center an object in both cameras, wherever the object may be, and a user can readily see where the robot is looking. The cameras give binocular visual information for e.g. use with virtual reality goggles, or for face recognition and computer vision processing. For sound generation, the iHome loudspeaker is used. For sound acquisition we use a Sony mini stereo microphone. These peripherals plug into a standard computer or e.g. Beagle Bones device for the developer to use in any way he or she would normally use USB cameras and speakers and microphones. The arms are also built from erector-set style parts. The device follows the proportions of an adult human. The elaborate system with two arms and Kinect weighs about 13 pounds. As depicted, the system can be set on a desktop or can be mounted on a standard microphone stand. Please visit: www.fluidbase.com to see movies and details, for options on how to get a system, and to join the discussion and development forum.

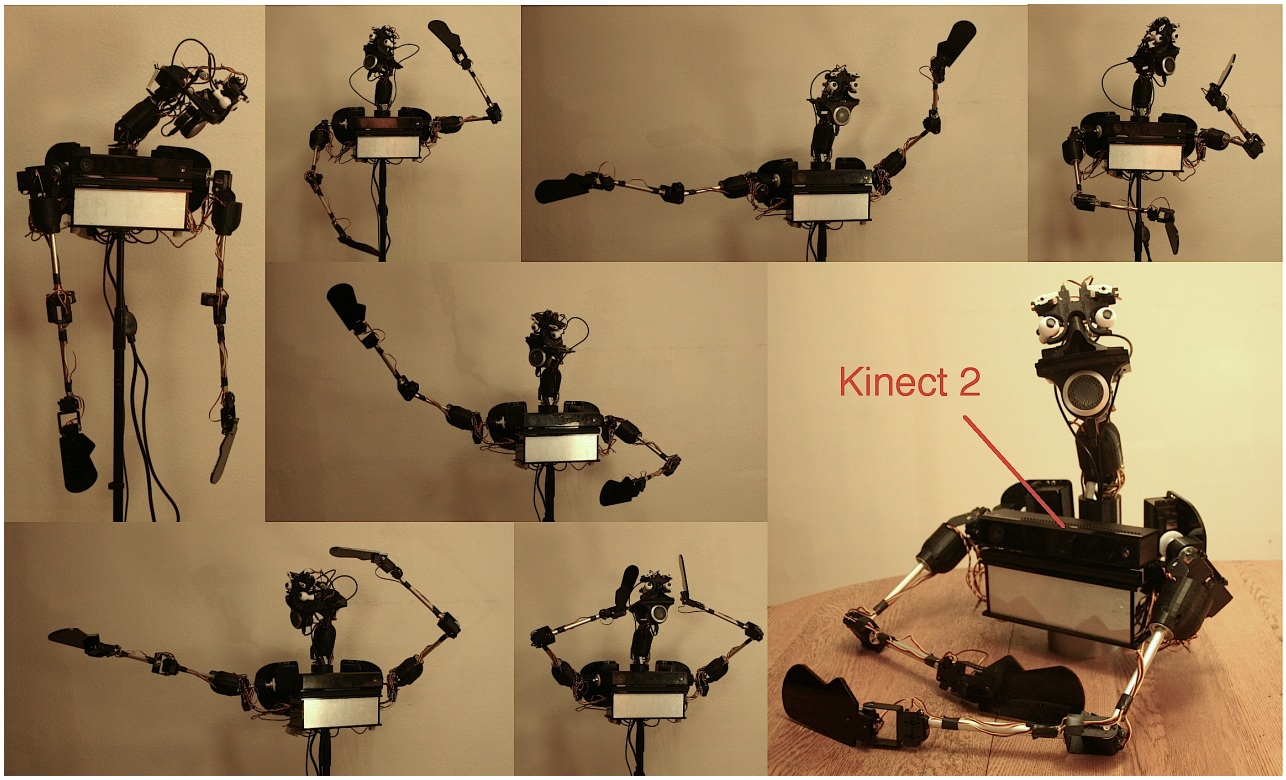

Figure 2: *illustrating the 9 degrees of freedom of the common system: (1) head rotate, (2) neck pan, (3) neck tilt, (4) head tilt, (5) eye-cams tilt, (6 & 7) left & right eye-cams pan, (8 & 9) left and right expressive eyebrows (optional). The full system includes two arms and an aluminum base wide enough to support an Xbox One Kinect V2 sensor system. Each arm has seven degrees of freedom: (1) wrist bend, (2) wrist rotate, (3) elbow bend, (4) elbow rotate, (5) arm elevate, (6) arm rotate, (7) shoulder pan.*

3. Software

Use of a device involves deciding how to control its motors. The widely used and open Robot Operating System (ROS) [2] provides a standard set of libraries and tools for building robot applications and is easily paired with the SSC32 motor control board that we use [3]. However, using ROS may be both more advanced and more limiting than a speech engineer may find useful. In light of this, we provide two other simple options for controlling motors. The first is a short Matlab script, and the second is some C/C++ source code that may be used to send commands to the SSC32. These two options are intended to get a developer quickly and easily up and running in building applications for their device from scratch. There are numerous ways to control motors, even over the internet (ROS is one) while the cameras, speaker, and microphone may be used by other software, such as Skype.

We also provide a framework for an open-source software project and user forum that will allow speech and robotics researchers and engineers, as well as hobbyists, to share their work and aid each other in problem solving. We eagerly seek feedback from the community in this endeavor. The idea is that developers may write individual components that integrate with other components written by other developers in a structured environment. We choose to use Matlab as a primary environment because Matlab already has many available tools (e.g. face detection and face recognition packages, advanced signal processing tools, an ROS toolkit, a package for working with the Microsoft Kinect, sound localization and separation tools, etc..) Furthermore, Matlab is a high-level environment that is accessible to mainstream linguists, psychologists, and students and non-expert computer programmers. Because there is a community that prefers not to use commercial tools such as Matlab, we also provide a C/C++ option (with ROS).

We anticipate that an open-access low-cost hardware platform is essential in forming a community of robot speech interaction developers. By introducing such a platform, we hope that a community will grow, and that such a community will define standards and generate a shared software base that will open new frontiers in interactive robotics development. We assume that progress will build on ideas and conventions being established in work with conversational avatars, such as Perception Markup Language [4], and Behavior Markup Language [5], as well as standards and conventions from the robotics community, such as ROS [2]. However, community feedback is needed. It is an open and exciting question how developers might use this platform and in what directions related speech software will grow. See [6] for more.

4. References

[1] http://www.lynxmotion.com
[2] http://www.ros.org/
[3] https://github.com/smd-ros-devel/lynxmotion_ssc32
[4] Scherer, S., Marsella, S., Stratou, G., Xu, Y., Morbini, F., (2012) Perception markup language: towards a standard representation of perceived non-verbal behaviors. In proc. of IVA. *Springer*
[5] Kopp, S., Krenn, B., Marsella, S., Marshall, N.A., Pelachuad, P., Pirker, H., Thorisson, K., Vihajlmsson, H. (2007) Toward a common framework for multimodal generation: the behavior markup language. In proc. IVA. *Springer*
[6] http://www.fluidbase.com